Titanic Dataset Data Cleaning and Preparation

Problem Statement

The task is to preprocess the Titanic dataset to prepare it for machine learning modeling. This involves handling missing values, removing outliers, converting categorical data into numeric format, and standardizing numerical values.

Solution Overview

1. **Loading and Inspecting the Dataset:**
    - Load the dataset into a pandas DataFrame.
    - Inspect the data for missing values, potential errors, and outliers.
2. **Data Cleaning:**
    - Handle missing values by filling them with appropriate imputation methods.
    - Create new features if necessary and drop irrelevant columns.
    - Identify and remove outliers.
3. **Data Transformation:**
    - Convert categorical data into numeric format using one-hot encoding.
    - Normalize or standardize the numerical values to ensure they are on a similar scale.

Challenges and Resolutions

1. **Missing Values:**
    - **Challenge:** The 'Age', 'Cabin', and 'Embarked' columns had missing values.
    - **Resolution:**
        - Filled missing 'Age' values with the median.
        - Created a new feature 'HasCabin' indicating whether a passenger had a cabin, and dropped the 'Cabin' column.
        - Filled missing 'Embarked' values with the mode.
2. **Outliers:**
    - **Challenge:** The 'Fare' column had outliers that could skew the analysis.
    - **Resolution:** Used the Interquartile Range (IQR) method to identify and remove outliers.
3. **Categorical Data:**
    - **Challenge:** Machine learning models require numeric input, but 'Sex' and 'Embarked' are categorical.
    - **Resolution:** Applied one-hot encoding to convert these categorical columns into numeric format.

4. **Normalization:**
    - **Challenge:** Different scales in the numerical data ('Age' and 'Fare') could affect the model performance.
    - **Resolution:** Standardized the 'Age' and 'Fare' columns to have a mean of 0 and a standard deviation of 1.

Instructions

1. Environment Setup:

To run the provided code, you need to have Python installed with the following libraries:

- pandas
- numpy
- matplotlib
- seaborn
- scikit-learn

You can install the required libraries using pip:

```
pip install pandas numpy matplotlib seaborn scikit-learn
```

2. Load and Inspect the Dataset:

```
import pandas as pd

# Load the dataset
file_path = '/mnt/data/train.csv'
df = pd.read_csv(file_path)

# Display the first few rows of the dataset
df.head()

# Inspect the data
df.info()
```

3. Data Cleaning:

```
# Handle missing values
df['Age'].fillna(df['Age'].median(), inplace=True)
df['HasCabin'] = df['Cabin'].notnull().astype(int)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
df.drop('Cabin', axis=1, inplace=True)

# Remove outliers in 'Fare'
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)
IQR = Q3 - Q1
```

```
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df = df[(df['Fare'] >= lower_bound) & (df['Fare'] <= upper_bound)]
```

4. Data Transformation:

```
# Convert categorical data into numeric format using one-hot encoding
df = pd.get_dummies(df, columns=['Embarked', 'Sex'], drop_first=True)

# Normalize or standardize the numerical values
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])

# Display the first few rows of the transformed dataset
df.head()
```

5. Verify the Transformations:

```
# Check for any remaining missing values
print("Missing Values after Imputation:\n", df.isnull().sum())

# Display summary statistics after standardization
df.describe()
```

Formatting and Readability

Headings and Subheadings:

- Use clear and descriptive headings for each section (e.g., "Problem Statement," "Solution Overview").
- Use subheadings for detailed steps within each section (e.g., "Loading and Inspecting the Dataset," "Data Cleaning").

Bullet Points and Diagrams:

- Use bullet points to list challenges, resolutions, and instructions clearly.
- Include diagrams where necessary to illustrate data distributions, outlier detection, and transformation processes.

Diagrams

Boxplot for Outliers:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 6))
```

```
sns.boxplot(x=df['Fare'])
plt.title('Boxplot of Fare')
plt.show()
```

Summary Statistics Before and After Standardization:

```
# Before Standardization
df[['Age', 'Fare']].describe()

# After Standardization
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])
df[['Age', 'Fare']].describe()
```

Conclusion

By following these steps, you'll have a clean and well-prepared dataset ready for machine learning modeling. The documented process ensures that each transformation is justified and reproducible, making the dataset suitable for various machine learning algorithms.